Lecture 4

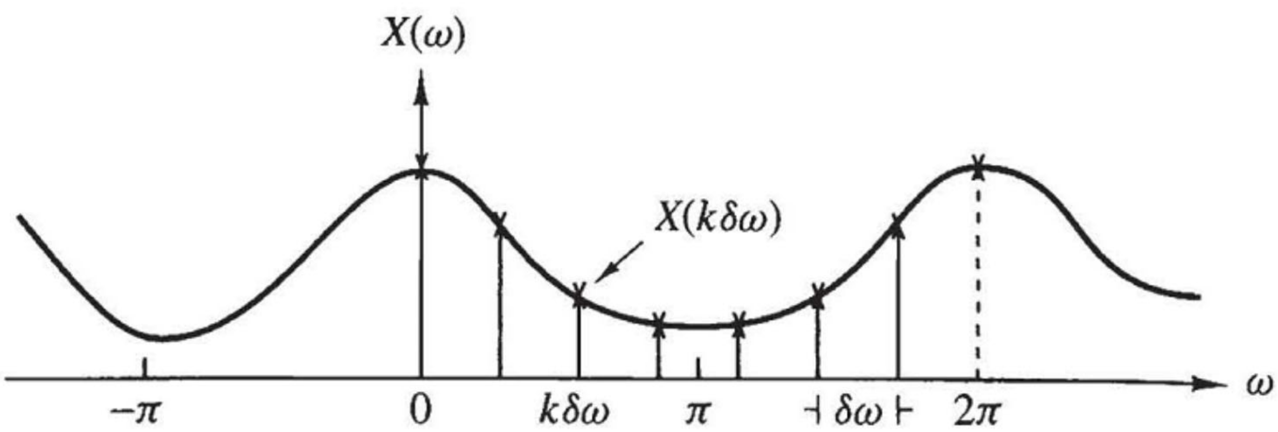
Discrete Fourier Transform (DFT) and its implementation: Fast Fourier Transform (FFT)

Frequency Domain Sampling

- Consider an aperiodic discrete-time signal x[n] with Fourier Transform

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n}$$

- Assume that we take samples of the spectrum spaced $\delta\omega$



Frequency Domain Sampling

- Let: $\omega = 2\pi k/N$, then:

$$X\left(\frac{2\pi}{N}k\right) = \sum_{n=-\infty}^{\infty} x(n)e^{-j2\pi kn/N}, \qquad k = 0, 1, \ldots, N-1$$

Sub-dividing the summation, we have:

$$X\left(\frac{2\pi}{N}k\right) = \cdots + \sum_{n=-N}^{-1} x(n)e^{-j2\pi kn/N} + \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}$$

$$+ \sum_{n=N}^{2N-1} x(n)e^{-j2\pi kn/N} + \cdots$$

$$= \sum^{\infty} \sum^{lN+N-1} x(n)e^{-j2\pi kn/N}$$

Frequency Domain Sampling

If we change the index in the inner summation from n to $n-lN$ and interchange the order of the summation, we obtain the result

$$X\left(\frac{2\pi}{N}k\right) = \sum_{n=0}^{N-1}\left[\sum_{l=-\infty}^{\infty} x(n-lN)\right] e^{-j2\pi kn/N}$$

for $k = 0, 1, 2, \ldots, N-1$.

The signal

$$x_p(n) = \sum_{l=-\infty}^{\infty} x(n-lN)$$

is clearly periodic with fundamental period N. Consequently, it can be expanded in a Fourier series as

$$x_p(n) = \sum_{k=0}^{N-1} c_k e^{j2\pi kn/N}, \qquad n = 0, 1, \ldots, N-1$$

DFT

- The coefficients are:

$$c_k = \frac{1}{N}\sum_{n=0}^{N-1} x_p(n)e^{-j2\pi kn/N}, \qquad k = 0, 1, \ldots, N-1$$

- It is easy to see that:

$$c_k = \frac{1}{N}X\left(\frac{2\pi}{N}k\right), \qquad k = 0, 1, \ldots, N-1$$

- Therefore,

$$x_p(n) = \frac{1}{N}\sum_{k=0}^{N-1} X\left(\frac{2\pi}{N}k\right)e^{j2\pi kn/N}, \qquad n = 0, 1, \ldots, N-1$$

- So, $x_p(n)$ can be recovered from Samples of $X(\omega)$

if there is no aliasing in the time domain,

$$x(n) = x_p(n), \qquad 0 \le n \le N-1$$

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X\left(\frac{2\pi}{N}k\right)e^{j2\pi kn/N}, \qquad 0 \le n \le N-1$$

DFT Pair

- So, we have DFT and Inverse DFT defined as:

$$\text{DFT:}\quad X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}, \qquad k = 0, 1, 2, \ldots, N-1$$

$$\text{IDFT:}\quad x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N}, \qquad n = 0, 1, 2, \ldots, N-1$$

DFT: Example

A finite-duration sequence of length L is given as

$$x(n) = \begin{cases} 1, & 0 \le n \le L-1 \\ 0, & \text{otherwise} \end{cases}$$

Determine the N-point DFT of this sequence for $N \ge L$.

Solution. The Fourier transform of this sequence is

$$X(\omega) = \sum_{n=0}^{L-1} x(n)e^{-j\omega n}$$

$$= \sum_{n=0}^{L-1} e^{-j\omega n} = \frac{1 - e^{-j\omega L}}{1 - e^{-j\omega}} = \frac{\sin(\omega L/2)}{\sin(\omega/2)} e^{-j\omega(L-1)/2}$$

The magnitude and phase of $X(\omega)$ are illustrated in Fig. 7.1.5 for $L = 10$. The N-point DFT of $x(n)$ is simply $X(\omega)$ evaluated at the set of N equally spaced frequencies $\omega_k = 2\pi k/N$, $k = 0, 1, \ldots, N-1$. Hence

$$X(k) = \frac{1 - e^{-j2\pi kL/N}}{1 - e^{-j2\pi k/N}}, \qquad k = 0, 1, \ldots, N-1$$

$$= \frac{\sin(\pi kL/N)}{\sin(\pi k/N)} e^{-j\pi k(L-1)/N}$$

DFT: Example

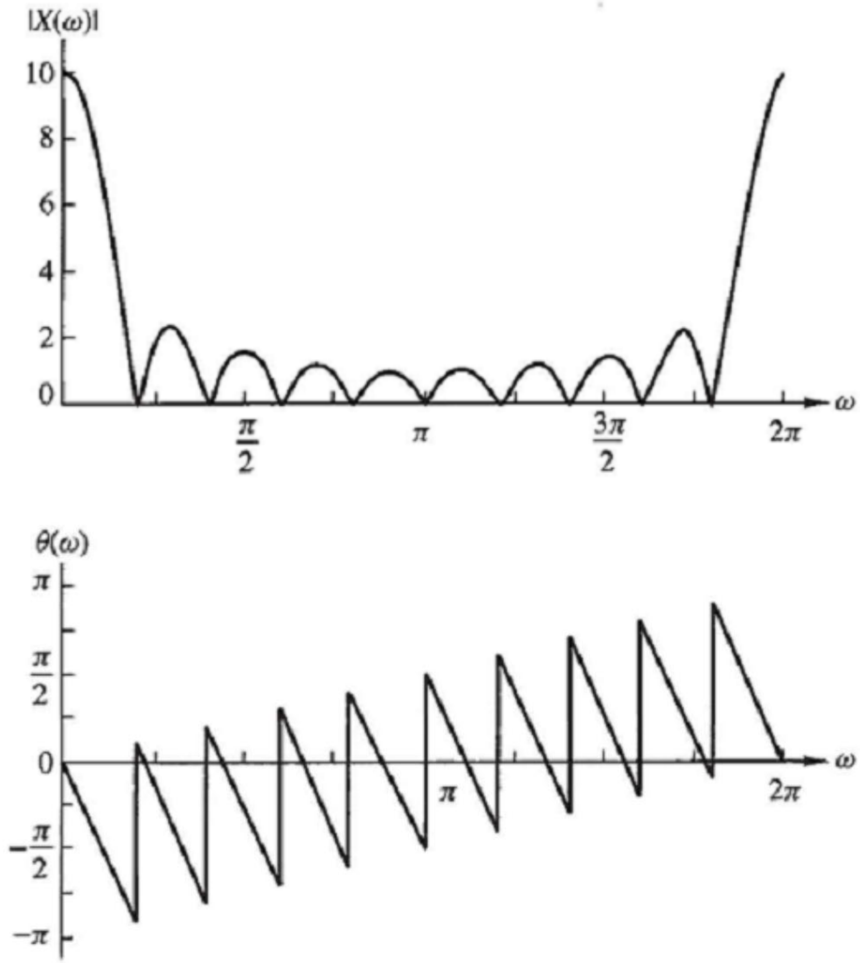


Figure 7.1.5
Magnitude and phase characteristics of the Fourier transform for signal in Example 7.1.2.

DFT as a Linear Transformation

- Let

$$W_N = e^{-j2\pi/N}$$

- Then,
-

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \qquad k = 0, 1, \ldots, N-1$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn}, \qquad n = 0, 1, \ldots, N-1$$

DFT as a Linear Transformation

- Define

$$\mathbf{x}_N = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}, \qquad \mathbf{X}_N = \begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix}$$

$$\mathbf{W}_N = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N & W_N^2 & \cdots & W_N^{N-1} \\ & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix}$$

- Then, the DFT and IDFT can be expressed as:
- DFT

$$\mathbf{X}_N = \mathbf{W}_N \mathbf{x}_N$$

- IDFT

$$\mathbf{x}_N = \mathbf{W}_N^{-1} \mathbf{X}_N = \frac{1}{N} \mathbf{W}_N^* \mathbf{X}_N$$

Note that

$$\mathbf{W}_N^{-1} = \frac{1}{N} \mathbf{W}_N^*$$

$$\mathbf{W}_N \mathbf{W}_N^* = N \mathbf{I}_N$$

Example

Compute the DFT of the four-point sequence

$$x(n) = (0 \quad 1 \quad 2 \quad 3)$$

Solution. The first step is to determine the matrix $\mathbf{W}_4$. By exploiting the periodicity property of $\mathbf{W}_4$ and the symmetry property $W_N^{k+N/2} = -W_N^k$

$$\mathbf{W}_4 = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W_4^1 & W_4^2 & W_4^3 \\ 1 & W_4^2 & W_4^0 & W_4^2 \\ 1 & W_4^3 & W_4^2 & W_4^1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

Then

$$\mathbf{X}_4 = \mathbf{W}_4\mathbf{x}_4 = \begin{bmatrix} 6 \\ -2+2j \\ -2 \\ -2-2j \end{bmatrix}$$

Relationship Between DFT and z-Transform

Let us consider

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n}$$

with an ROC that includes the unit circle.

$$X(k) \equiv X(z)|_{z=e^{j2\pi nk/N}}, \qquad k = 0, 1, \ldots, N-1$$

$$= \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi nk/N}$$

then

$$X(k) \equiv X(z)|_{z=e^{j2\pi nk/N}}, \qquad k = 0, 1, \ldots, N-1$$

$$= \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi nk/N}$$

Relationship Between DFT and z-transform

If the sequence $x(n)$ has a finite duration of length N or less, the sequence can be recovered from its N-point DFT.

$$X(z) = \sum_{n=0}^{N-1} x(n) z^{-n} = \sum_{n=0}^{N-1} \left[\frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N} \right] z^{-n}$$

$$X(z) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \sum_{n=0}^{N-1} \left(e^{j2\pi k/N} z^{-1} \right)^{n}$$

$$X(z) = \frac{1 - z^{-N}}{N} \sum_{k=0}^{N-1} \frac{X(k)}{1 - e^{j2\pi k/N} z^{-1}}$$

When evaluated on the unit circle,

$$X(\omega) = \frac{1 - e^{-j\omega N}}{N} \sum_{k=0}^{N-1} \frac{X(k)}{1 - e^{-j(\omega - 2\pi k/N)}}$$

Properties of DFT

$$\text{DFT:} \quad X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \qquad k = 0, 1, \ldots, N-1$$

$$\text{IDFT:} \quad x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn}, \qquad n = 0, 1, \ldots, N-1$$

where W_N is defined as

$$W_N = e^{-j2\pi/N}$$

Properties of DFT

Periodicity. If $x(n)$ and X(k) are an N-point DFT pair, then

$$x(n+N) = x(n) \qquad \text{for all } n$$

$$X(k+N) = X(k) \qquad \text{for all } k$$

Linearity. If

$$x_1(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X_1(k)$$

and

$$x_2(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X_2(k)$$

then for any real-valued or complex-valued constants a_1 and a_2,

$$a_1x_1(n) + a_2x_2(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} a_1X_1(k) + a_2X_2(k)$$

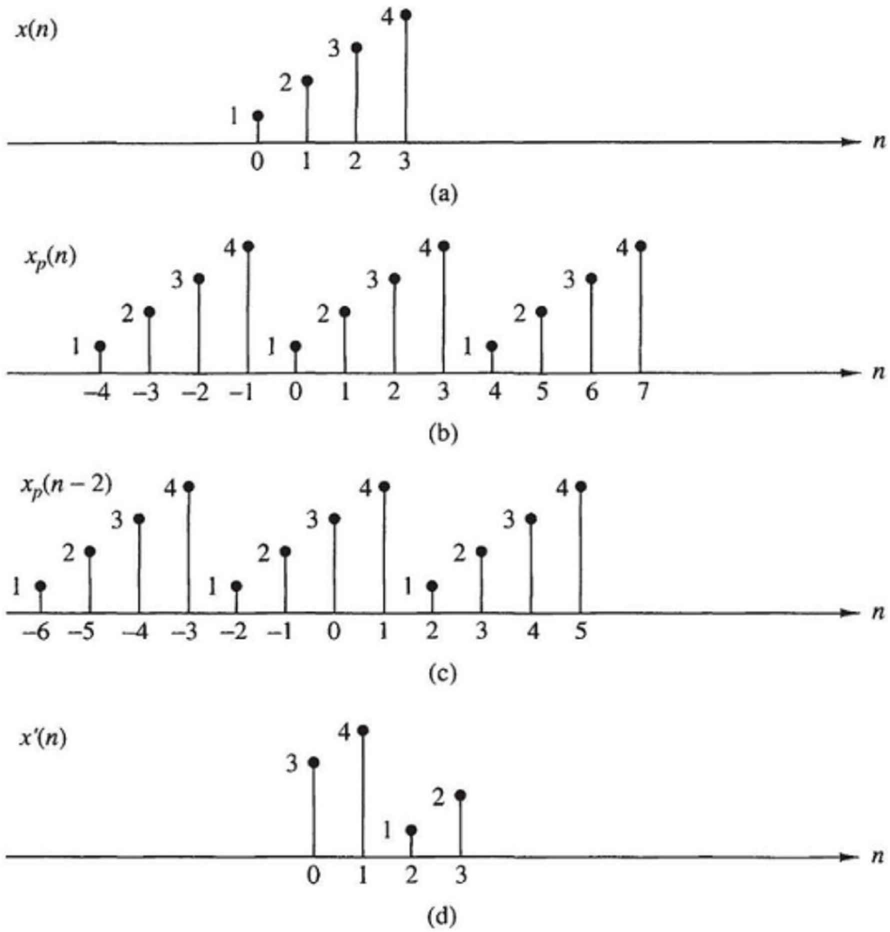
Properties of DFT: Circular Symmetries

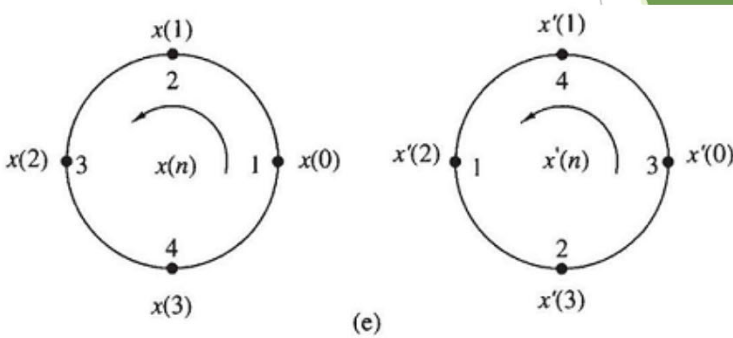
$$x_p(n) = \sum_{l=-\infty}^{\infty} x(n - lN)$$

Shifting $x_p(n)$ by k, we get,

$$x'_p(n) = x_p(n-k) = \sum_{l=-\infty}^{\infty} x(n - k - lN)$$

Properties of DFT: Circular Symmetries





Circular shift of a sequence.

$$x'(n) = x(n-k, \text{modulo } N)$$
$$\equiv x((n-k))_N$$

Properties of DFT: Circular Convolution

Take,
$$X_1(k) = \sum_{n=0}^{N-1} x_1(n) e^{-j2\pi nk/N}, \qquad k = 0, 1, \ldots, N-1$$

and
$$X_2(k) = \sum_{n=0}^{N-1} x_2(n) e^{-j2\pi nk/N}, \qquad k = 0, 1, \ldots, N-1$$

Multiply the two to get,
$$X_3(k) = X_1(k) X_2(k), \qquad k = 0, 1, \ldots, N-1$$

IDFT is,
$$x_3(m) = \frac{1}{N} \sum_{k=0}^{N-1} X_3(k) e^{j2\pi km/N}$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} X_1(k) X_2(k) e^{j2\pi km/N}$$

Properties of DFT: Circular Convolution

Suppose that we substitute for $X_1(k)$ and $X_2(k)$ we obtain

$$x_3(m) = \frac{1}{N}\sum_{k=0}^{N-1}\left[\sum_{n=0}^{N-1} x_1(n)e^{-j2\pi kn/N}\right]\left[\sum_{l=0}^{N-1} x_2(l)e^{-j2\pi kl/N}\right]e^{j2\pi km/N}$$

$$= \frac{1}{N}\sum_{n=0}^{N-1} x_1(n)\sum_{l=0}^{N-1} x_2(l)\left[\sum_{k=0}^{N-1} e^{j2\pi k(m-n-l)/N}\right]$$

The inner sum in the brackets has the form

$$\sum_{k=0}^{N-1} a^k = \begin{cases} N, & a = 1 \\ \dfrac{1-a^N}{1-a}, & a \neq 1 \end{cases}$$

where a is defined as

$$a = e^{j2\pi(m-n-l)/N}$$

Circular Convolution

We observe that $a = 1$ when $m - n - l$ is a multiple of N. On the other hand, $a^N = 1$ for any value of $a \neq 0$. Consequently,

$$\sum_{k=0}^{N-1} a^k = \begin{cases} N, & l = m - n + pN = ((m-n))_N, \quad p \text{ an integer} \\ 0, & \text{otherwise} \end{cases}$$

If we substitute the result in $x_3(m)$, we obtain the desired expression for $x_3(m)$ in the form

$$x_3(m) = \sum_{n=0}^{N-1} x_1(n) x_2((m-n))_N, \qquad m = 0, 1, \ldots, N-1$$

The expression has the form of a convolution sum. However, it is not the ordinary linear convolution that was introduced in Chapter 2, which relates the output sequence $y(n)$ of a linear system to the input sequence $x(n)$ and the impulse response $h(n)$. Instead, the convolution sum in involves the index $((m-n))_N$ and is called *circular convolution*. Thus we conclude that multiplication of the DFTs of two sequences is equivalent to the circular convolution of the two sequences in the time domain.

Properties of the DFT

Property	Time Domain	Frequency Domain
Notation	$x(n),\ y(n)$	$X(k),\ Y(k)$
Periodicity	$x(n) = x(n+N)$	$X(k) = X(k+N)$
Linearity	$a_1x_1(n) + a_2x_2(n)$	$a_1X_1(k) + a_2X_2(k)$
Time reversal	$x(N-n)$	$X(N-k)$
Circular time shift	$x((n-l))_N$	$X(k)e^{-j2\pi kl/N}$
Circular frequency shift	$x(n)e^{j2\pi ln/N}$	$X((k-l))_N$
Complex conjugate	$x^*(n)$	$X^*(N-k)$
Circular convolution	$x_1(n) \circledN x_2(n)$	$X_1(k)X_2(k)$
Circular correlation	$x(n) \circledN y^*(-n)$	$X(k)Y^*(k)$
Multiplication of two sequences	$x_1(n)x_2(n)$	$\frac{1}{N}X_1(k) \circledN X_2(k)$
Parseval's theorem	$\sum_{n=0}^{N-1} x(n)y^*(n)$	$\frac{1}{N}\sum_{k=0}^{N-1} X(k)Y^*(k)$

Properties of DFT

- Special case of Parseval's Theorem,
- When $y(n) = x(n)$

$$\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2$$

- This means that it does not matter whether you compute energy in time-domain or frequency-domain.

Linear Filtering using DFT

Let, $x(n) = 0, \qquad n < 0 \text{ and } n \geq L$

$h(n) = 0, \qquad n < 0 \text{ and } n \geq M$

- The output y[n] will be,

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k)$$

With duration $L + M - 1$.

In frequency domain, we have,

$$Y(\omega) = X(\omega)H(\omega)$$

Linear Filtering using DFT: Example

Let,

$$Y(k) \equiv Y(\omega)|_{\omega=2\pi k/N}, \qquad k = 0, 1, \ldots, N-1$$

$$= X(\omega)H(\omega)|_{\omega=2\pi k/N}, \qquad k = 0, 1, \ldots, N-1$$

- Then,

$$Y(k) = X(k)H(k), \qquad k = 0, 1, \ldots, N-1$$

- Where

$\{X(k)\}$ and $\{H(k)\}$ are the N-point DFTs of the sequences $x(n)$ and $h(n)$

- Example: Find the response of $h(n) = \{\underset{\uparrow}{1}, 2, 3\}$ to $x(n) = \{\underset{\uparrow}{1}, 2, 2, 1\}$

Linear Filtering using DFT: Example

Let, N=L+M-1=4+3-1=6.

Then, $$X(k) = \sum_{n=0}^{7} x(n) e^{-j2\pi kn/8}$$

$$= 1 + 2e^{-j\pi k/4} + 2e^{-j\pi k/2} + e^{-j3\pi k/4}, \qquad k = 0, 1, \ldots, 7$$

- We have,

$$X(0) = 6, \qquad X(1) = \frac{2+\sqrt{2}}{2} - j\left(\frac{4+3\sqrt{2}}{2}\right)$$

$$X(2) = -1 - j, \qquad X(3) = \frac{2-\sqrt{2}}{2} + j\left(\frac{4-3\sqrt{2}}{2}\right)$$

$$X(4) = 0, \qquad X(5) = \frac{2-\sqrt{2}}{2} - j\left(\frac{4-3\sqrt{2}}{2}\right)$$

$$X(6) = -1 + j, \qquad X(7) = \frac{2+\sqrt{2}}{2} + j\left(\frac{4+3\sqrt{2}}{2}\right)$$

Linear Filtering using DFT: Example

For H[k],

$$H(k) = \sum_{n=0}^{7} h(n) e^{-j2\pi kn/8}$$

$$= 1 + 2e^{-j\pi k/4} + 3e^{-j\pi k/2}$$

- So,

$$H(0) = 6, \quad H(1) = 1 + \sqrt{2} - j\left(3 + \sqrt{2}\right), \quad H(2) = -2 - j2$$

$$H(3) = 1 - \sqrt{2} + j\left(3 - \sqrt{2}\right), \quad H(4) = 2$$

$$H(5) = 1 - \sqrt{2} - j\left(3 - \sqrt{2}\right), \quad H(6) = -2 + j2$$

$$H(7) = 1 + \sqrt{2} + j\left(3 + \sqrt{2}\right)$$

Linear Filtering using DFT: Example

Multiplying X[k] and H[k], we get

$$Y(0) = 36, \quad Y(1) = -14.07 - j17.48, \quad Y(2) = j4, \quad Y(3) = 0.07 + j0.515$$
$$Y(4) = 0, \quad Y(5) = 0.07 - j0.515, \quad Y(6) = -j4, \quad Y(7) = -14.07 + j17.48$$

Finally, the eight-point IDFT is

$$y(n) = \sum_{k=0}^{7} Y(k) e^{j2\pi kn/8}, \qquad n = 0, 1, \ldots, 7$$

This computation yields the result

$$y(n) = \{\underset{\uparrow}{1}, 4, 9, 11, 8, 3, 0, 0\}$$

We observe that the first six values of $y(n)$ constitute the set of desired output values. The last two values are zero because we used an eight-point DFT and IDFT, when, in fact, the minimum number of points required is six.

Aliasing in DFT

We needed N to be at least L+M-1. In the example, we had to have N at least 6 and we used N=8 and it worked. Now let's use N=4:

$$H(k) = \sum_{n=0}^{3} h(n) e^{-j2\pi kn/4}$$

$$H(k) = 1 + 2e^{-j\pi k/2} + 3e^{-jk\pi}, \qquad k = 0, 1, 2, 3$$

Hence

$$H(0) = 6, \qquad H(1) = -2 - j2, \qquad H(2) = 2, \qquad H(3) = -2 + j2$$

The four-point DFT of $x(n)$ is

$$X(k) = 1 + 2e^{-j\pi k/2} + 2e^{-j\pi k} + 1e^{-j3\pi k/2}, \qquad k = 0, 1, 2, 3$$

Hence

$$X(0) = 6, \qquad X(1) = -1 - j, \qquad X(2) = 0, \qquad X(3) = -1 + j$$

The product of these two four-point DFTs is

$$\hat{Y}(0) = 36, \qquad \hat{Y}(1) = j4, \qquad \hat{Y}(2) = 0, \qquad \hat{Y}(3) = -j4$$

Aliasing in DFT

The 4-point IDFT gives:

$$\hat{y}(n) = \frac{1}{4}\sum_{k=0}^{3}\hat{Y}(k)e^{j2\pi kn/4}, \qquad n = 0, 1, 2, 3$$

$$= \frac{1}{4}(36 + j4e^{j\pi n/2} - j4e^{j3\pi n/2})$$

- So, we have:

$$\hat{y}(n) = \{\underset{\uparrow}{9}, 7, 9, 11\}$$

- We see that y[4] is aliased with y[0] and y[5] is aliased with y[1]:

$$\hat{y}(0) = y(0) + y(4) = 9 \qquad \hat{y}(1) = y(1) + y(5) = 7$$

and the other two terms are ok:

$$\hat{y}(2) = y(2) = 9$$

$$\hat{y}(3) = y(3) = 11$$

Filtering of Long Data Sequences

Usually, we need to process very long sequences of data. So, we need to cut the sequence into a large number of blocks, each of a reasonably short length N and find N-point DFT's and combine them. There are tw wasy to do this:

1) Overlap-Save Method,
2) Overlap-Add Method.

Overlap-save method. In this method the size of the input data blocks is $N = L + M - 1$ and the DFTs and IDFT are of length N. Each data block consists of the last $M-1$ data points of the previous data block followed by L new data points to form a data sequence of length $N = L + M - 1$. An N-point DFT is computed for each data block. The impulse response of the FIR filter is increased in length by appending $L - 1$ zeros and an N-point DFT of the sequence is computed once and stored. The multiplication of the two N-point DFTs $\{H(k)\}$ and $\{X_m(k)\}$ for the mth block of data yields

$$\hat{Y}_m(k) = H(k)X_m(k), \qquad k = 0, 1, \ldots, N-1$$

Overlap-Save Method

Then the N-point IDFT yields the result

$$\hat{Y}_m(n) = \{\hat{y}_m(0)\hat{y}_m(1)\cdots\hat{y}_m(M-1)\hat{y}_m(M)\cdots\hat{y}_m(N-1)\}$$

Since the data record is of length N, the first $M-1$ points of $y_m(n)$ are corrupted by aliasing and must be discarded. The last L points of $y_m(n)$ are exactly the same as the result from linear convolution and, as a consequence,

$$\hat{y}_m(n) = y_m(n), n = M, M+1, \ldots, N-1$$

To avoid loss of data due to aliasing, the last $M-1$ points of each data record are saved and these points become the first $M-1$ data points of the subsequent record, as indicated above. To begin the processing, the first $M-1$ points of the first record are set to zero. Thus the blocks of data sequences are

$$x_1(n) = \{\underbrace{0, 0, \ldots, 0,}_{M-1 \text{ points}} x(0), x(1), \ldots, x(L-1)\}$$

Overlap-Save Method

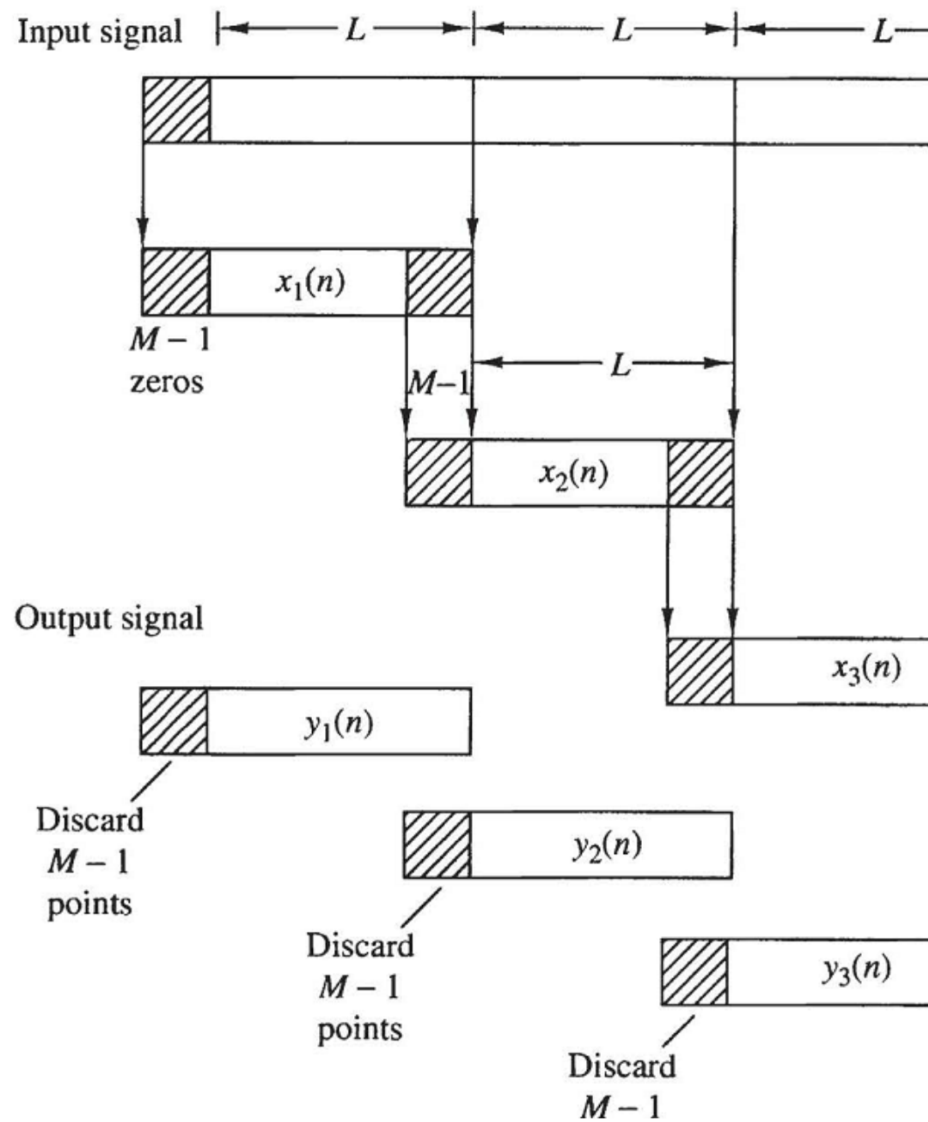
$$x_1(n) = \{\underbrace{0, 0, \ldots, 0,}_{M-1 \text{ points}} x(0), x(1), \ldots, x(L-1)\}$$

$$x_2(n) = \{\underbrace{x(L-M+1), \ldots, x(L-1),}_{M-1 \text{data points from } x_1(n)} \underbrace{x(L), \ldots, x(2L-1)}_{L \text{ new data points}}\}$$

$$x_3(n) = \{\underbrace{x(2L-M+1), \ldots, x(2L-1),}_{M-1 \text{ data points from } x_2(n)} \underbrace{x(2L), \ldots, x(3L-1)}_{L \text{ new data points}}\}$$

- and so on.

The first $M-1$ points are discarded due to aliasing and the remaining L points constitute the desired result from linear convolution.



Overlap-Add Method

In this method the size of the input data block is L points and the size of the DFTs and IDFT is $N = L+M-1$. To each data block we append $M-1$ zeros and compute the N-point DFT. Thus the data blocks may be represented as

$$x_1(n) = \{x(0), x(1), \ldots, x(L-1), \underbrace{0, 0, \ldots, 0}_{M-1 \text{ zeros}}\}$$

$$x_2(n) = \{x(L), x(L+1), \ldots, x(2L-1), \underbrace{0, 0, \ldots, 0}_{M-1 \text{ zeros}}\}$$

$$x_3(n) = \{x(2L), \ldots, x(3L-1), \underbrace{0, 0, \ldots, 0}_{M-1 \text{ zeros}}\}$$

and so on. The two N-point DFTs are multiplied together to form

$$Y_m(k) = H(k)X_m(k), \qquad k = 0, 1, \ldots, N-1$$

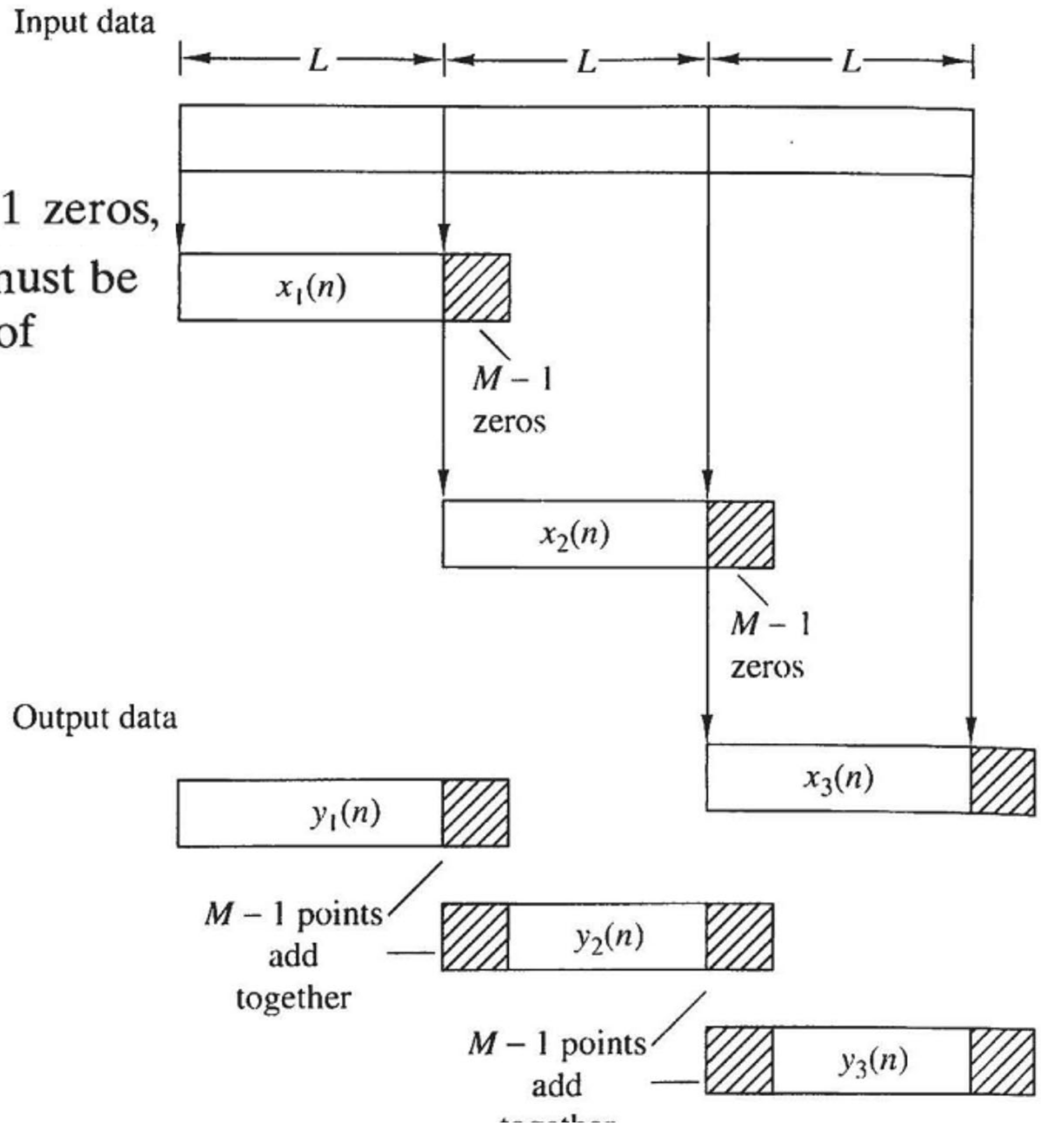
The IDFT yields data blocks of length N that are free of aliasing, since the size of the DFTs and IDFT is $N = L + M - 1$ and the sequences are increased to N-points by appending zeros to each block.

Overlap-Add Method

Since each data block is terminated with $M - 1$ zeros, the last $M - 1$ points from each output block must be overlapped and added to the first $M - 1$ points of

the succeeding block. Hence this method is called the overlap-add method. This overlapping and adding yields the output sequence

$$y(n) = \{y_1(0), y_1(1), \ldots, y_1(L-1), y_1(L) + y_2(0), y_1(L+1) + y_2(1), \ldots, y_1(N-1) + y_2(M-1), y_2(M), \ldots\}$$



Efficient Implementation of DFT

The DFT and IDFT are given by,

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \qquad 0 \le k \le N-1$$

and,

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk}, \qquad 0 \le n \le N-1$$

where, $W_N = e^{-j2\pi/N}$

W_N has the following properties:

Symmetry property: $W_N^{k+N/2} = -W_N^k$

Periodicity property: $W_N^{k+N} = W_N^k$

Direct Implementation of DFT

For a complex-valued sequence x[n] the DFT and IDFT are is given by,

$$X_R(k) = \sum_{n=0}^{N-1} \left[x_R(n) \cos \frac{2\pi kn}{N} + x_I(n) \sin \frac{2\pi kn}{N} \right]$$

$$X_I(k) = -\sum_{n=0}^{N-1} \left[x_R(n) \sin \frac{2\pi kn}{N} - x_I(n) \cos \frac{2\pi kn}{N} \right]$$

The direct computation requires:

1. $2N^2$ evaluations of trigonometric functions.
2. $4N^2$ real multiplications.
3. $4N(N-1)$ real additions.
4. A number of indexing and addressing operations.

Splitting of the sequence

To simplify the task, we split x[n] into two sequences one consisting of the values with odd index and the other consisting of even indexed values:

$$f_1(n) = x(2n)$$
$$f_2(n) = x(2n+1), \quad n = 0, 1, \ldots, \frac{N}{2} - 1$$

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \qquad k = 0, 1, \ldots, N-1$$

$$= \sum_{n \text{ even}} x(n) W_N^{kn} + \sum_{n \text{ odd}} x(n) W_N^{kn}$$

$$= \sum_{m=0}^{(N/2)-1} x(2m) W_N^{2mk} + \sum_{m=0}^{(N/2)-1} x(2m+1) W_N^{k(2m+1)}$$

Splitting of the sequence

But, $W_N^2 = W_{N/2}$

So,

$$X(k) = \sum_{m=0}^{(N/2)-1} f_1(m) W_{N/2}^{km} + W_N^k \sum_{m=0}^{(N/2)-1} f_2(m) W_{N/2}^{km}$$

$$= F_1(k) + W_N^k F_2(k), \qquad k = 0, 1, \ldots, N-1$$

where $F_1(k)$ and $F_2(k)$ are the $N/2$-point DFTs of $f_1(m)$ and $f_2(m)$,

Since $F_1(k)$ and $F_2(k)$ are periodic, with period $N/2$, we have $F_1(k+N/2) = F_1(k)$ and $F_2(k+N/2) = F_2(k)$. In addition, the factor $W_N^{k+N/2} = -W_N^k$.

$$X(k) = F_1(k) + W_N^k F_2(k), \qquad k = 0, 1, \ldots, \frac{N}{2} - 1$$

$$X\left(k + \frac{N}{2}\right) = F_1(k) - W_N^k F_2(k), \qquad k = 0, 1, \ldots, \frac{N}{2} - 1$$

Splitting of the sequence

We observe that the direct computation of $F_1(k)$ requires $(N/2)^2$ complex multiplications. The same applies to the computation of $F_2(k)$. Furthermore, there are $N/2$ additional complex multiplications required to compute $W_N^k F_2(k)$. Hence the computation of $X(k)$ requires $2(N/2)^2 + N/2 = N^2/2 + N/2$ complex multiplications. This first step results in a reduction of the number of multiplications from N^2 to $N^2/2 + N/2$, which is about a factor of 2 for N large.

We define:

$$G_1(k) = F_1(k) \quad , \quad k = 0, 1, \ldots, \frac{N}{2} - 1$$

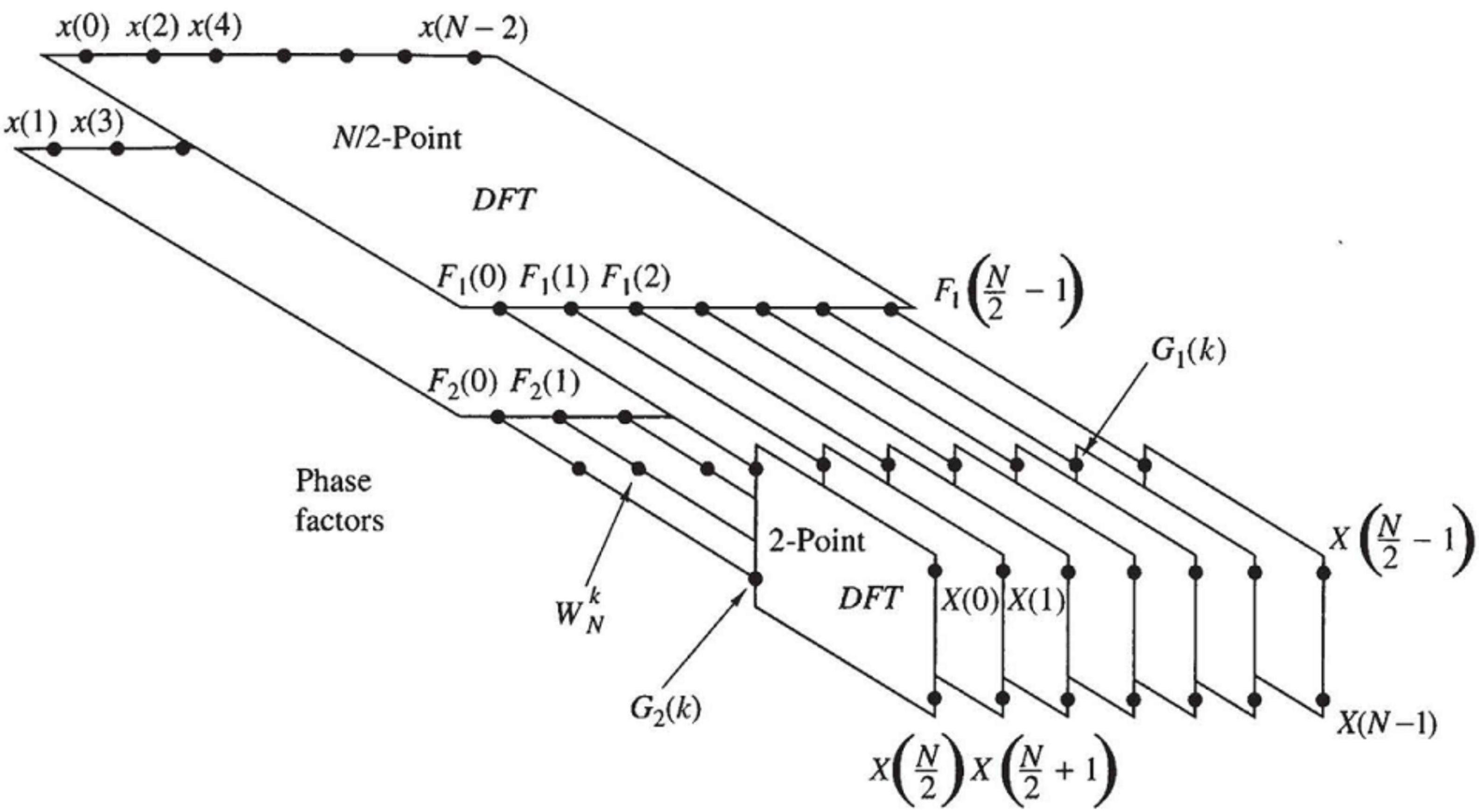
$$G_2(k) = W_N^k F_2(k), \quad k = 0, 1, \ldots, \frac{N}{2} - 1$$

The DFT can be expressed as:

$$X(k) = G_1(k) + G_2(k), \quad k = 0, 1, \ldots, \frac{N}{2} - 1$$

$$X(k + \frac{N}{2}) = G_1(k) - G_2(k), \quad k = 0, 1, \ldots, \frac{N}{2} - 1$$

Splitting of the sequence



Splitting of the sequence

We can now further divide

$$v_{11}(n) = f_1(2n), \qquad n = 0, 1, \ldots, \frac{N}{4} - 1$$

$$v_{12}(n) = f_1(2n+1), \quad n = 0, 1, \ldots, \frac{N}{4} - 1$$

and:

$$v_{21}(n) = f_2(2n), \qquad n = 0, 1, \ldots, \frac{N}{4} - 1$$

$$v_{22}(n) = f_2(2n+1), \quad n = 0, 1, \ldots, \frac{N}{4} - 1$$

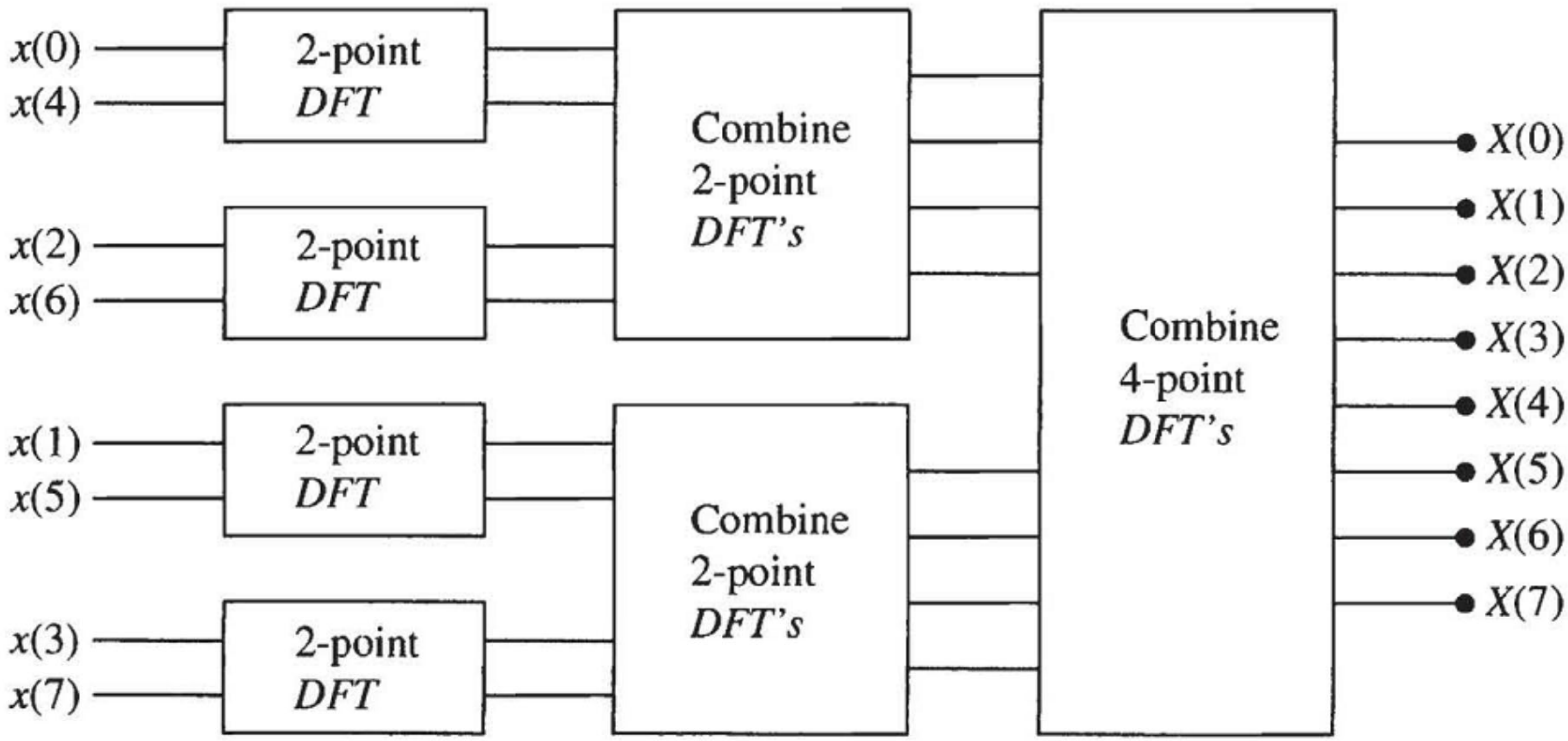
$$F_1(k) = V_{11}(k) + W_{N/2}^k V_{12}(k), \quad k = 0, 1, \ldots, \frac{N}{4} - 1$$

$$F_1\left(k + \frac{N}{4}\right) = V_{11}(k) - W_{N/2}^k V_{12}(k), \quad k = 0, 1, \ldots, \frac{N}{4} - 1$$

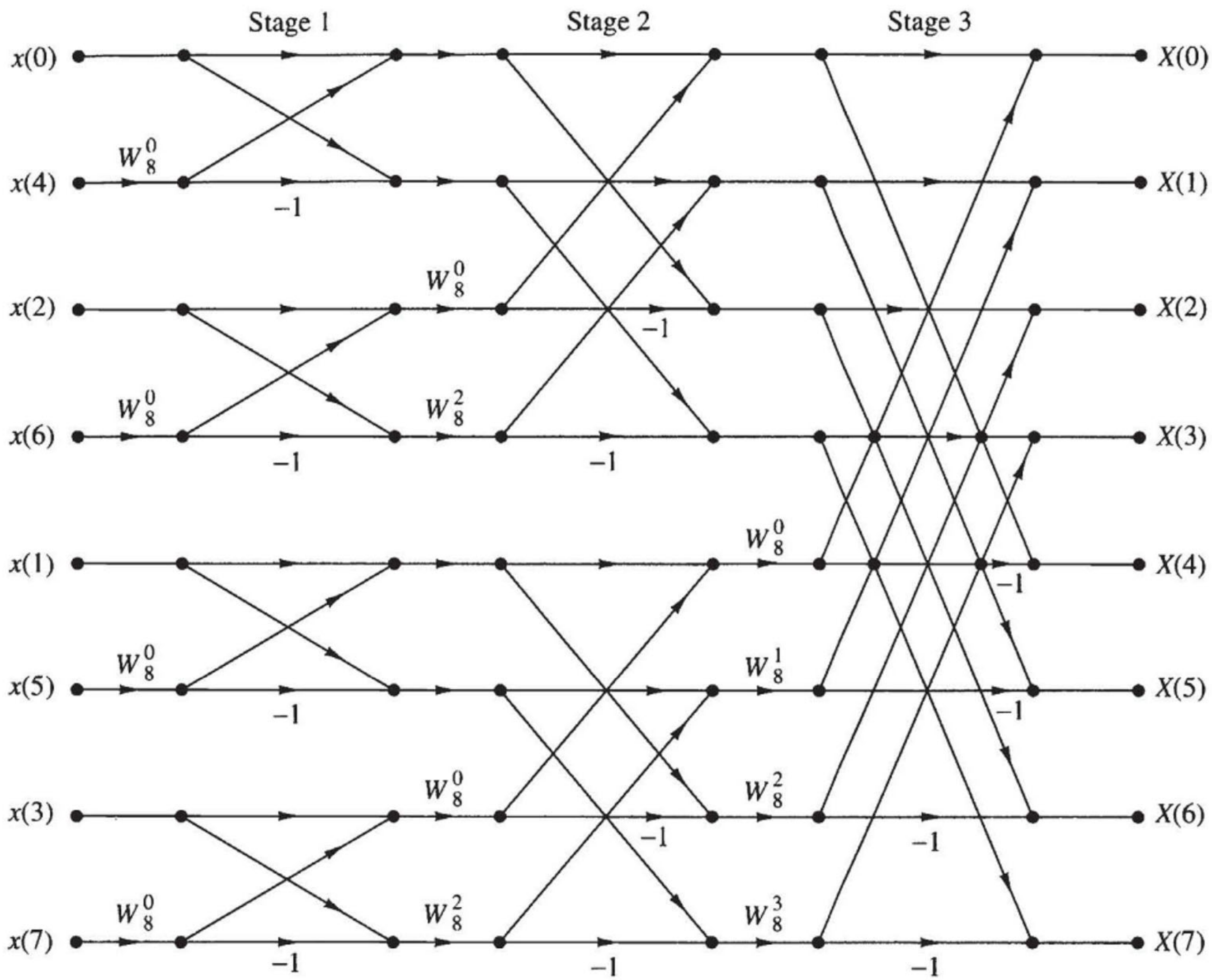
$$F_2(k) = V_{21}(k) + W_{N/2}^k V_{22}(k), \quad k = 0, 1, \ldots, \frac{N}{4} - 1$$

$$F_2\left(k + \frac{N}{4}\right) = V_{21}(k) - W_{N/2}^k V_{22}(k), \quad k = 0, \ldots, \frac{N}{4} - 1$$

Fast Fourier Transform (FFT)

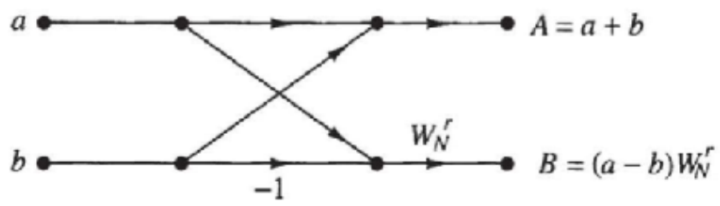


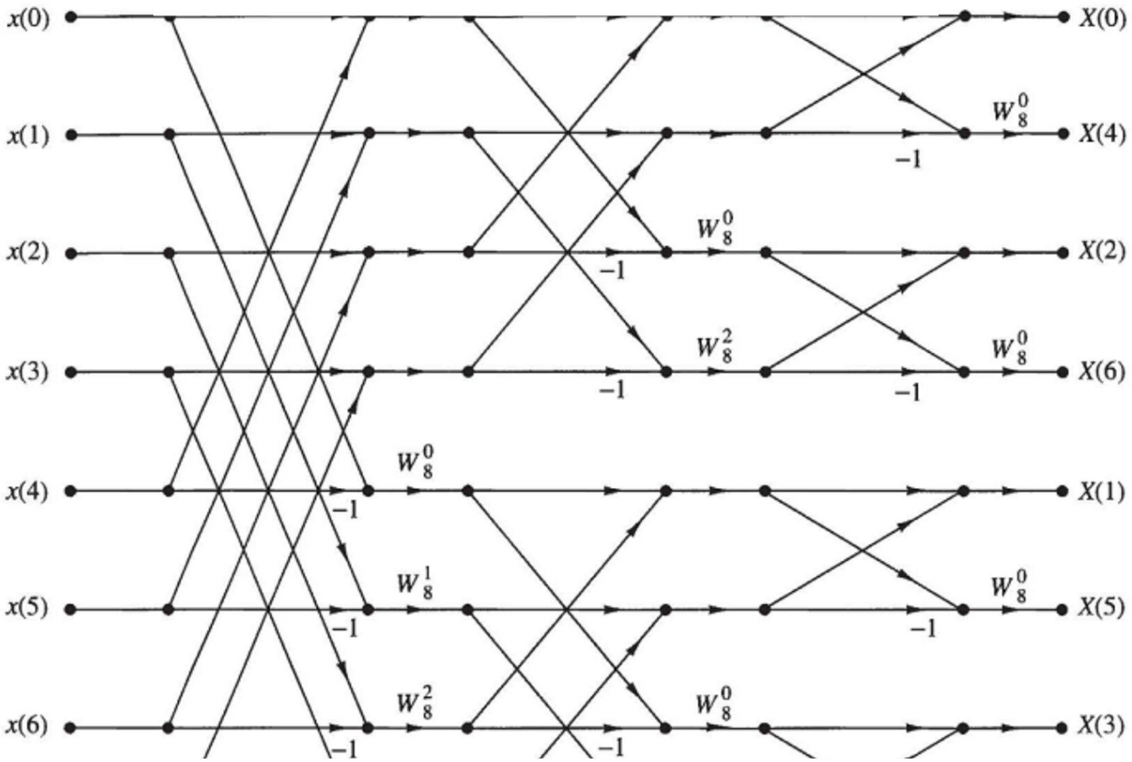
Three stages in the computation of an $N = 8$-point DFT.


Stage 1
Stage 2
Stage 3
x(0)
x(4)
x(2)
x(6)
x(1)
x(5)
x(3)
x(7)
X(0)
X(1)
X(2)
X(3)
X(4)
X(5)
X(6)
X(7)
W_8^0
W_8^1
W_8^2
W_8^3
−1

Fast Fourier Transform (FFT)

Basic Butterfly:





Fast Fourier Transform (FFT)

Comparison of Computational Complexity for the Direct Computation of the DFT Versus the FFT Algorithm

Number of Points, N	Complex Multiplications in Direct Computation, N^2	Complex Multiplications in FFT Algorithm, $(N/2)\log_2 N$	Speed Improvement Factor
4	16	4	4.0
8	64	12	5.3
16	256	32	8.0
32	1,024	80	12.8
64	4,096	192	21.3
128	16,384	448	36.6
256	65,536	1,024	64.0
512	262,144	2,304	113.8
1,024	1,048,576	5,120	204.8